International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 02 | February - 2023 Impact Factor: 7.185 ISSN: 2582-3930

PATTERN RECOGNITION FROM IMAGES AND VIDEOS USING MACHINE LEARNING

M.SARAVANAKUMAR
Department of COMPUTER APPLICATION
MADURAI KAMARAJ UNIVERSITY
MADURAI, INDIA

CHAPTER 1

ABSTRACT

Pattern recognition is a data analysis method that uses machine learning algorithms to automatically recognize patterns and regularities in data. This data can be anything from text and images to sounds or other definable qualities. Pattern recognition is supervised or unsupervised classification. Among the various frameworks in which pattern recognition has been traditionally formulated, the statistical approach structural and new has been most intensively studied and used in practice. More recently, neural network techniques and methods imported from statistical learning theory have deserved increasing attention. There are three main types of pattern recognition, dependent on the mechanism used for classifying the input data. Those types are: statistical, structural (or syntactic), and neural. Based on the type of processed data, it can be divided into image, sound, voice, and speech pattern recognition.

INTRODUCTION

The ability to analyze facial expressions plays a major role in non-verbal communication. If a someone only analyzes what a person's mouth says and ignores what the person's face says, then we can only have a part of the story. Humans were the only ones who could distinguish between expressions but not anymore, with advancing technology our computers can learn how to detect emotions as well. This report is a guide to facial expression recognition software using OpenCV, Keras, Tensorflow and CNN, by implementing a program in Python it has become possible to build an algorithm that performs detection, extraction, and evaluation of these facial expressions for automatic recognition of human emotion in real- time. The main

features of the face are considered for the detection of facial expressions. To determine the different emotions, the variations in each of the main features are used. To detect and classify different classes of emotions, machine learning algorithms are used by training different sets of images. This paper discusses a real-time emotion classification of a facial expression into one among the seven universal human expressions: Anger, Disgust, Fear, Happy, Neutral, Sad, Surprise by the implementation of a real-time vision system that can classify emotions.

Age and gender that are the two key facial attributes play a foundational role in social interactions, making age and gender estimation from one face image a crucial task in intelligent applications, like access control, human-computer interaction, enforcement, marketing intelligence and visual surveillance. The basic aim of this project is to develop an algorithm that estimates age and gender of a person correctly. One of the most widely used techniques is haar cascade. In this project I propose a model which can predict the gender of a person with the assistance of Haar Cascade. The model trained the classifier with different male and female images as positive and negative images. Different facial features are extracted. With the assistance of Haar Cascade classifier will determine whether the input image is male or female. I made use of Deep Convolution neural network. It works efficiently even with limited data. For the age approximation task, the project makes use of caffedeep learning framework. Caffe provides expressive architecture, extensible code. Caffe can process over 60M photos per day. This makes it one of the fastest convent implementation available. Facial expressions are more explanatory in situations where words fail, like a shock or a surprise. Moreover, lying through spoken words is more difficult to detect compared to faking expressions. Statistically non-verbal forms of communication make up about 66% of the total communication. This makes it very essential to study, use and analyze facial expressions.

Techniques for face acquisition

Essentially, the process of face recognition is performed in two steps. The first involves feature extraction and selection and the second is the classification of objects. Later developments introduced varying technologies to the procedure. Some of the most notable include the following techniques:

1. Traditional

Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of

the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches: geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances. Some classify these algorithms into two broad categories: holistic and feature-based models. The former attempts to recognize the face in its entirety while the feature-based subdivided into components such as according to features and analyze each as well as its spatial location with respect to other features.

Popular recognition algorithms include principal component analysis using Eigen faces, linear discriminant analysis, elastic bunch graph matching using the Fisher face algorithm, the hidden Markov model, the multilinear subspace learning using tensor representation, and the neuronal motivated dynamic link matching.

2. 3-Dimensional recognition

Three-dimensional face recognition technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin.

One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of face recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery. The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same CMOS chip—each sensor captures a different part of the spectrum.

Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the Technion applied tools from metric geometry to treat expressions as isometries. A new method is to introduce a way to capture a 3D picture by using three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, second one to the side, and third one at an angle. All these

cameras will work together so it can track a subject's face in real time and be able to face detect and recognize.

3. Skin texture analysis

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called Skin Texture Analysis, turns the unique lines, patterns, and spots apparent in a person's skin into a mathematical space.

Surface Texture Analysis works much the same way as facial recognition does. A picture is taken of a patch posada distinguish any lines, pores and the actual skin texture. It can identify the contrast between identical pairs, which are not yet possible using facial recognition software alone. Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent.

4. Facial recognition combining different techniques

As every method has its advantages and disadvantages, technology companies have amalgamated the traditional, 3D recognition and Skin Textual Analysis, to create recognition systems that have higher rates of success. Combined techniques have an advantage over other systems. It is relatively insensitive to changes in expression, including blinking, frowning or smiling and has the ability to compensate for mustache or beard growth and the appearance of eyeglasses. The system is also uniform with respect to race and gender.

5. Thermal cameras

A different form of taking input data for face recognition is by using thermal cameras, by this procedure the cameras will only detect the shape of the head and it will ignore the subject accessories such as glasses, hats, or makeup. Unlike conventional cameras, thermal cameras can capture facial imagery even in low-light and nighttime conditions without using a flash and exposing the position of the camera. However, a problem with using thermal pictures for face recognition is that the database for face recognition is limited. Diego Socolinsky and Andrea Selinger (2004) research the use of thermal face recognition in real life and operation sceneries, and at the same time build a new database of thermal face images. The research uses low-sensitive, low-resolution ferroelectric sensors that are capable of acquiring long-wave thermal infrared (LWIR). The results show that a fusion of LWIR and regular visual cameras has greater results in outdoor probes. Indoor results show that visual has a 97.05% accuracy, while LWIR has 93.93%, and the fusion has 98.40%, however on the outdoor proves visual has 67.06%, LWIR 83.03%, and fusion has 89.02%. The study used

240 subjects over a period of 10 weeks to create a new database. The data was collected on sunny, rainy, and cloudy days.

In 2018, researchers from the U.S. Army Research Laboratory (ARL) developed a technique that would allow them to match facial imagery obtained using a thermal camera with those in databases that were captured using a conventional camera. This approach utilized artificial intelligence and machine learning to allow researchers to visibly compare conventional and thermal facial imagery. And it is known as a cross-spectrum synthesis method due to how it bridges facial recognition from two different imaging modalities, this method synthesizes a single image by analyzing multiple facial regions and details.

ARL scientists have noted that the approach works by combining global information (i.e. features across the entire face) with local information (i.e. features regarding the eyes, nose, and mouth). In addition to enhancing the discriminability of the synthesized image, the facial recognition system can be used to transform a thermal face signature into a refined visible image of a face. According to performance tests conducted at ARL, researchers found that the multi-region cross-spectrum synthesis model demonstrated a performance improvement of about 30% over baseline methods and about 5% over state-of-the-art methods. It has also been tested for landmark detection for thermal images.

6. Face Recognition Working Steps

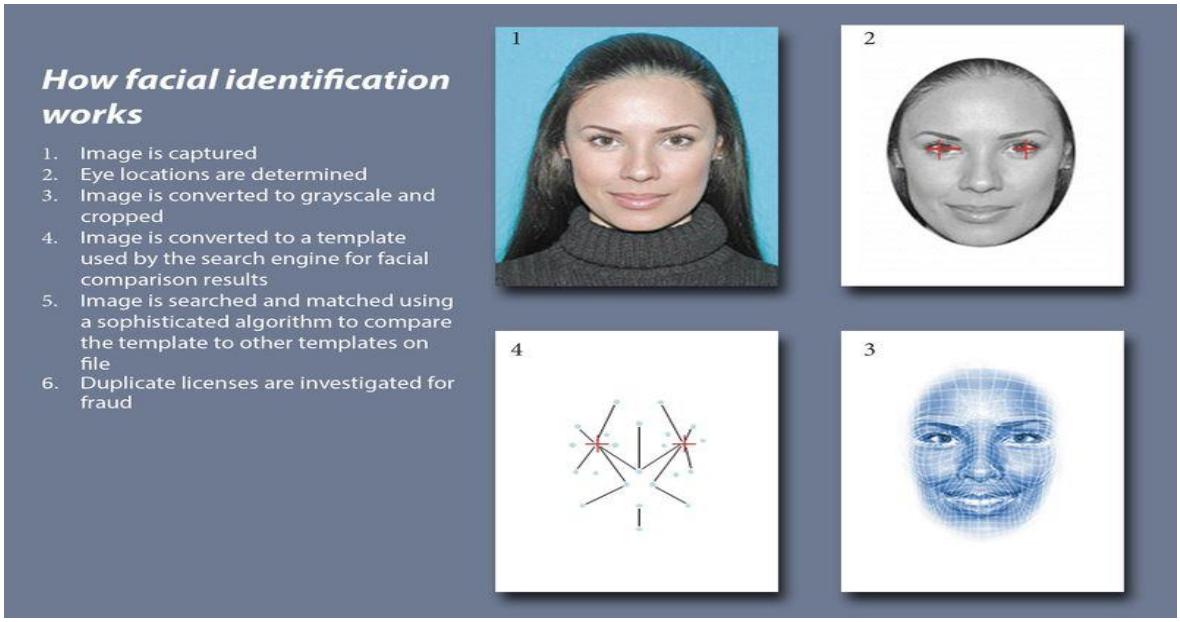


Figure 1.1 Facial identification works

Face recognition systems use computer algorithms to pick out specific, distinctive details about a person's face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. The data about a particular face is often called a face template and is distinct from a photograph because it's designed to only include certain details that can be used to distinguish one face from another.

Some face recognition systems, instead of positively identifying an unknown person, are designed to calculate a probability match score between the unknown person and specific face templates stored in the database. These systems will offer up several potential matches, ranked in order of likelihood of correct identification, instead of just returning a single result.

Face recognition systems vary in their ability to identify people under challenging conditions such as poor lighting, low quality image resolution, and suboptimal angle of view (such as in a photograph taken from above looking down on an unknown person).

Emotion Recognition

Emotion Recognition has three different parts:

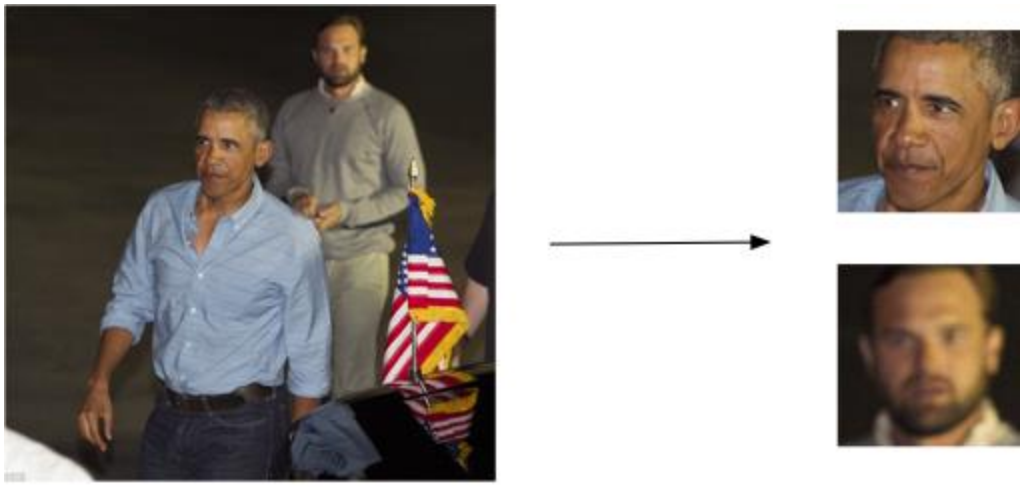
1. Facial Detection — Ability to detect the location of face in any input image or frame. The output is the bounding box coordinates of the detected faces

2. Facial Recognition — Compare multiple faces together to identify which faces belong to the same person. This is done by comparing face embedding vectors

3. Emotion Detection — Classifying the emotion on the face as happy, angry, sad, neutral, surprise, disgust or fea

Facial Detection

Facial detection is the first part of our pipeline. We have used the python library Face Recognition that we found easy to install and very accurate in detecting faces. This library scans the input image and returns the bounding box coordinates of all detected faces as shown below:



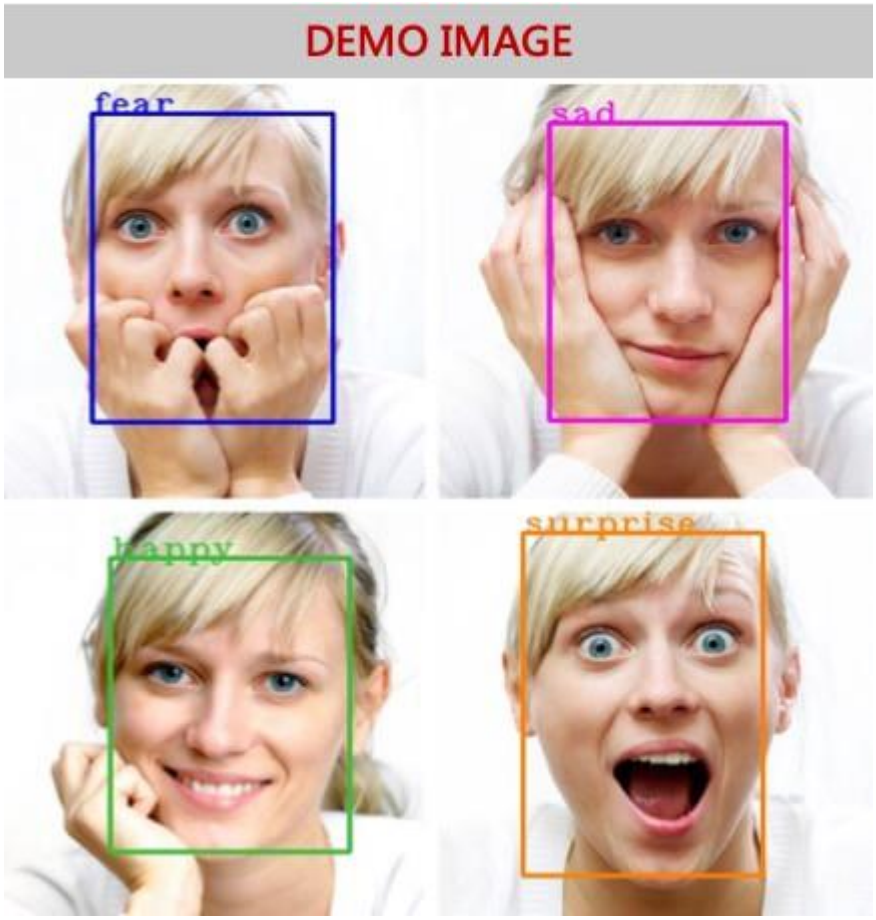
Input Output

Facial Recognition

Facial Recognition verifies if two faces are same. The use of facial recognition is huge in security, biometrics, entertainment, personal safety, etc. The same python library face recognition used for face detection can also be used for face recognition. Our testing showed it had good performance. Given two faces match, they can be matched with each other giving the result as True or False. The steps involved in facial recognition are

- Find face in an image
- Analyze facial feature
- Compare features for the 2 input faces
- Returns True if matched or else False.

Emotion Detection



Humans are used to taking in non verbal cues from facial emotions. Now computers are also getting better to reading emotions. So how do we detect emotions in an image? We have used an open source data set — Face Emotion Recognition (FER) from Kaggle and built a CNN to detect emotions. The emotions can be classified into 7 classes — happy, sad, fear, disgust, angry, neutral and surprise.

Model — We built a 6 layered Convolutional Neural Network (CNN) in Keras and use image augmentations to improve model performance.

BASIC TERMINOLOGIES

Face Detection:

Face detection is to determine that a certain picture contains a face we need to be able to define the general structure of face. Luckily human faces do not greatly differ from each other; we all have noses, eyes, foreheads, chins and mouths; and all of these compose the general structure of a face. It is a concept of two-class classification: face versus nonface. Face detection can be regarded as a specific case of objectclass detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class.

Face Identification:

In this the system compares the given individual to all the other individuals in the database and gives a ranked list of matches.

Face Verification:

In this the system compares the given individual with who that individual says they are and gives a yes or no decision.

Facial Expressions:

Facial expression is one or more motions or positions of the muscles beneath the skin of the face. These movements express the emotional state of the person to observers. It is a form of non-verbal communication. It plays a communicative role in interpersonal relations. The common ones are:

CHAPTER 2

LITERATURE REVIEW

2.1 EXISTING SYSTEM

[1] Bartlett, Marian, et al. "Data mining spontaneous facial behavior with automatic expression coding." Verbal and Nonverbal Features of Human-Human and Human Machine Interaction. Springer, Berlin, Heidelberg, 2008. 1-20.

Nowadays, deep learning techniques know a big success in various fields including computer vision. Indeed, a convolutional neural networks (CNN) model can be trained to analyze images and identify face emotion. In this paper, we create a system that recognizes students' emotions from their faces. Our system consists of three phases: face detection using Haar Cascades, normalization and emotion recognition using CNN on FER 2013 database with seven types of expressions. Obtained results show that face emotion recognition is feasible in education, consequently, it can help teachers to modify their presentation according to the students' emotions.

[2] Raghuvanshi, Arushi, and Vivek Choksi. "Facial expression recognition with convolutional neural networks." CS231n Course Projects 362 (2016).

With the transition from laboratory-controlled to daunting in-the-wild conditions of facial Emotion recognition (FER) and the recent popularity of deep learning strategies in various fields, deep neural networks[1], [2] have rapidly been leveraged to train discriminatory representations for automated FER. The FER will allow us to recognize the Emotion of the human face that is a major blow to recent technological development. Recent FER programmers are typically concentrating on two critical issues: overfitting due to lack of appropriate training evidence and emotion-related differences such as lighting, head posture and identification bias. It encourages to improve the other innovations, such as incorporating FER into the robotic device in order to provide the robot feelings.

[3] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI '15, (New York, NY, USA), pp. 435–442, ACM, 2015.

As it is known, sentiments influence information processing, attitude formation, and decision making to a great extent in real-world scenarios. Several recent efforts have been published about FER or facial expression recognition, however, due to the diversity of human faces and fluctuations in pictures, reliable and robust FER systems remain a challenge. Till date, every study and work has proposed either a single network or an ensemble model. The accuracy of ensemble models is higher, but they were associated with many models and datasets and a few tweaked datasets to improve the accuracy, increasing the computing complexity. While the majority of research in this field focuses on improving accuracy, this study utilizes the proposed model to a real-world scenario in which a person's face contains a mix of emotions, and a single-label sentiment can be highly noisy in such situations.

[4] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in Proc. ACM International Conference on Multimodal Interaction (ICMI), November 2015.

Facial emotion recognition is one of the promptly developing branches within the machine learning domain. In this paper, we are presenting our model based on Convolutional Neural Networks, which is trained on Cohn-Kanade and RAVDESS datasets. The proposed model gets satisfactory results in detecting macro facial emotions on the aforementioned datasets.

[5] B. Kim, J. Roh, S. Dong, and S. Lee, "Hierarchical committee of deep convolutional neural networks for robust facial expression recognition," Journal on Multimodal User Interfaces, pp. 1–17, 2016.

An emotion recognition system can be built by utilizing the benefits of deep learning and different applications such as feedback analysis, face unlocking etc. can be implemented with good accuracy. The main focus of this work is to create a Deep Convolutional Neural Network (DCNN) model that classifies 5

different human facial emotions. The model is trained, tested and validated using the manually collected image dataset.

[6] Fan, Yin, et al. "Video-based emotion recognition using CNN-RNN and C3D hybrid networks." Proceedings of the 18th ACM International Conference on Multimodal Interaction. 2016.

In recent years, an increased number of intelligent systems are using facial emotion recognition to improve human interaction. These systems cause constant changes in their operation based on the emotion of humans. In this paper, we propose an architecture based on the convolutional neural network (CNN) for the facial recognition of emotions. In the implementation, we use the Facial Expression Recognition 2013 dataset (FER2013). Through behavioral analysis, we show how different emotions seem to be sensitive to different parts of the face.

Limitation in Existing System

Human face may be a storehouse of various information about personal characteristics, including identity, emotional expression, gender, age, etc. the looks of face is affected considerably by aging. This plays a significant role in nonverbal communication between humans. Age and gender, two key facial attributes, play a really foundational role in social interactions, making age and gender estimation from one face image a very important task in machine learning applications, like access control, human-computer interaction, law enforcement, marketing intelligence and visual surveillance. Automatic gender classification and age detection may be a fundamental task in computer vision, which has recently attracted immense attention. It plays a very important role in an exceedingly wide selection of the real-world applications like targeted advertisement, forensic science, visual surveillance, content-based searching, human computer interaction systems, etc. for instance I are able to use this method to display advertisement supported different gender and different age bracket.

Disadvantage of Existing System

- The manual process in detection of age will be a risky task which won't implement in a real-time.
- In voting system, the age detection helps in knowing the age group which can't be detect in manual work.
- In car driving, the analyzing of age in automatic manner helps in allowing the person to drive. This

implementation can't execute in manual concept.

- In car driving, the camera attached in the vehicle helps in detecting the age and thus allowing the corresponding person to drive or not can be predicted. Only age of above 18 will be allowed.

PROPOSED SYSTEM

Recognition of gender is kind of difficult when the image is captured from far distance by using haar-like features. For this problem I have used simple but effective idea. I applied cascaded method. The project uses ROI (Region of interest) as my face. I returned the ROI image to classifier. In this project, I tried to detect the female face. I trained my haar cascade classifier by 500 female and 500 male images. I used frontal face images to train that included external features like hairstyle, makeup, accessories such as earrings and glasses. The object that this project is trying to detect is positive in xml training. The CNN algorithm helps in identifying the age and gender by extracting the data from the input image which get compared with the dataset and thus prediction will be executed. The output label will be placed at the resultant image.

Advantage of Proposed System

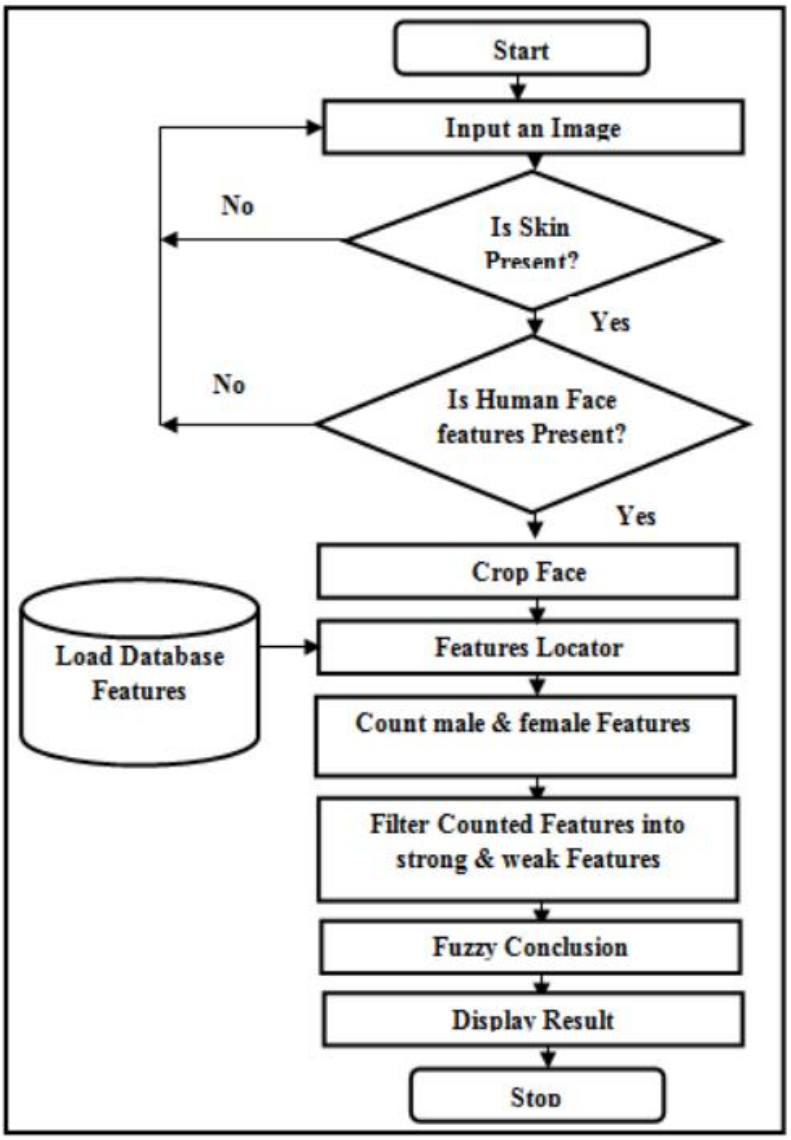
- In voting system, the age detection helps in knowing the age group which can be detect in computerized manner. The camera at the entrance will predict the person age group and allow 18+ age person for voting.
- In car driving, the camera attached in the vehicle helps in detecting the age and thus allowing the corresponding person to drive or not can be predicted. Only age of above 18 will be allowed.
- The door opening as per gender can be automated. The camera predicts the person gender and as per the prediction, the corresponding door will be open.

CHAPTER 3

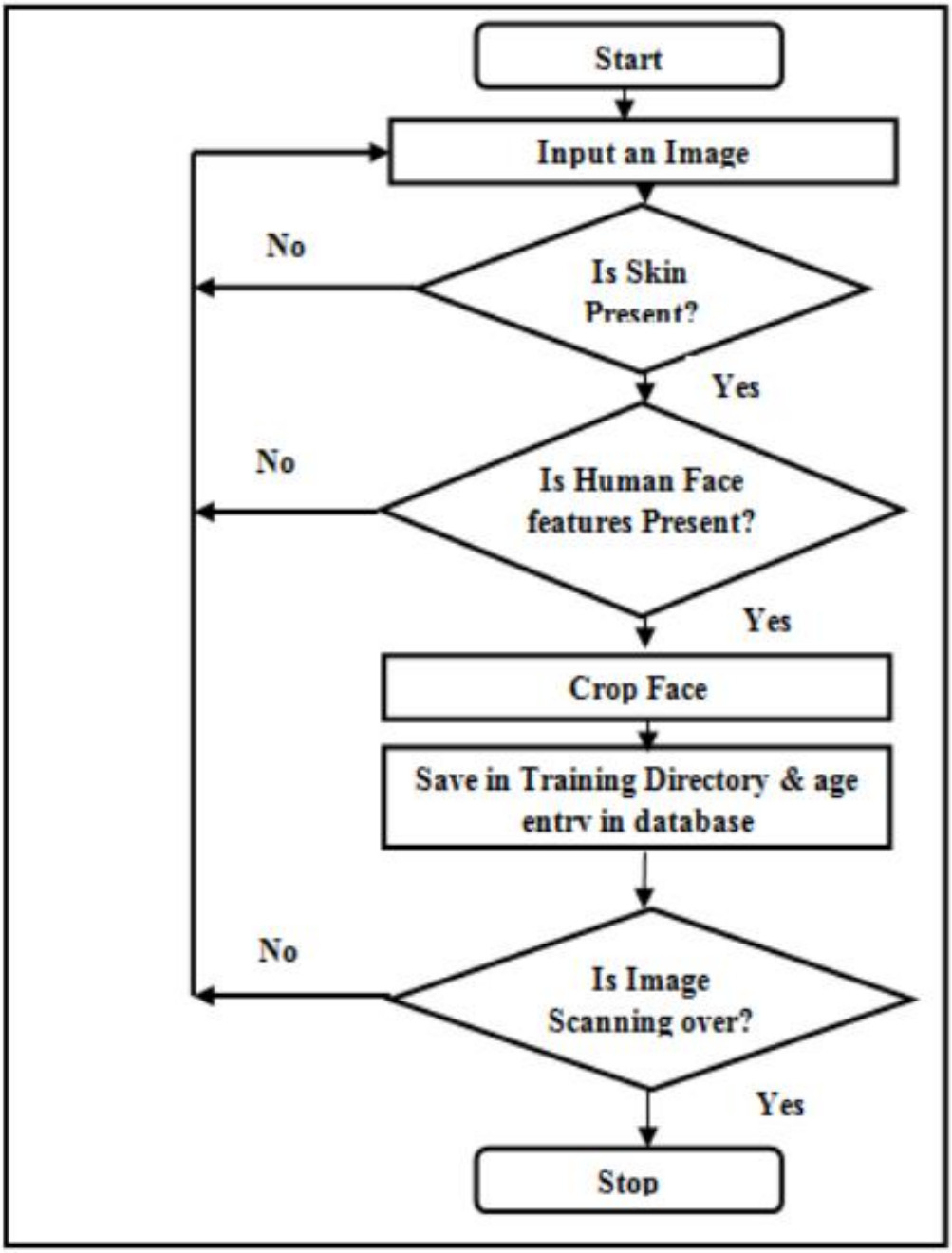
SYSTEM DESIGN

Data Flow Diagram

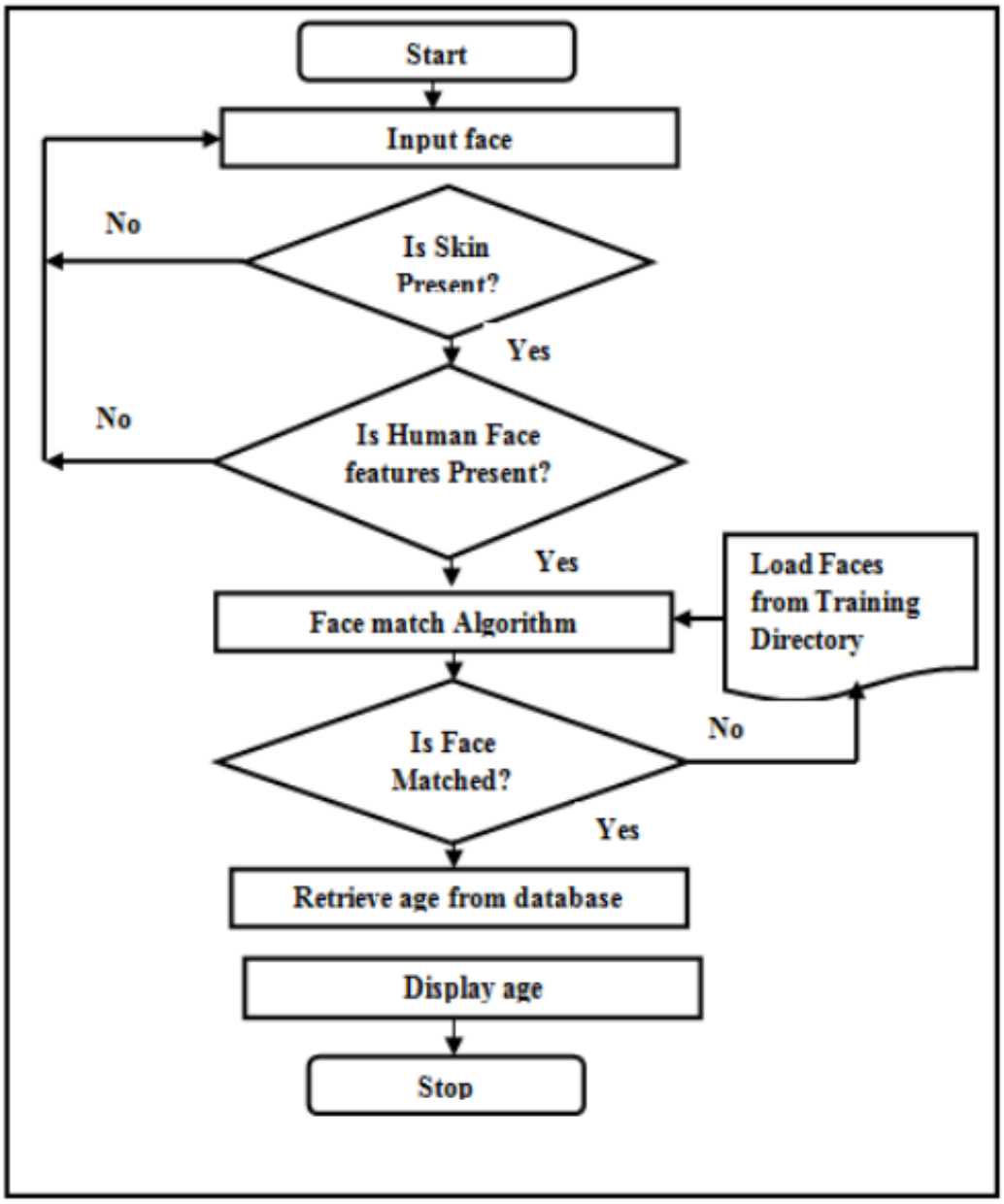
Level - 0



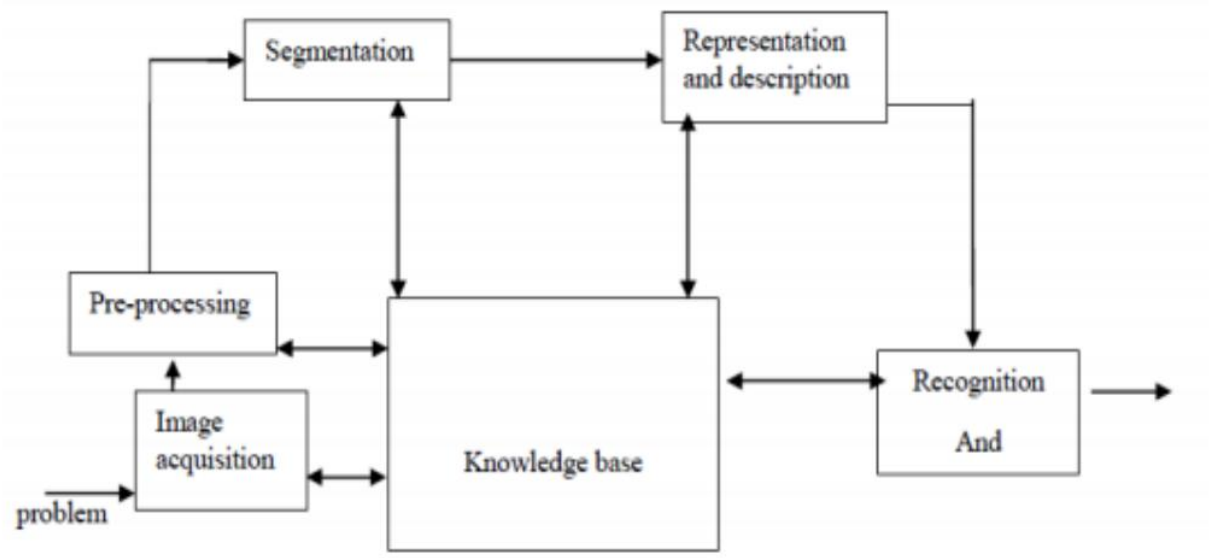
Level – 1


Start
Input an Image
No
Is Skin Present?
Yes
No
Is Human Face features Present?
Yes
Crop Face
Save in Training Directory & age entry in database
No
Is Image Scanning over?
Yes
Stop

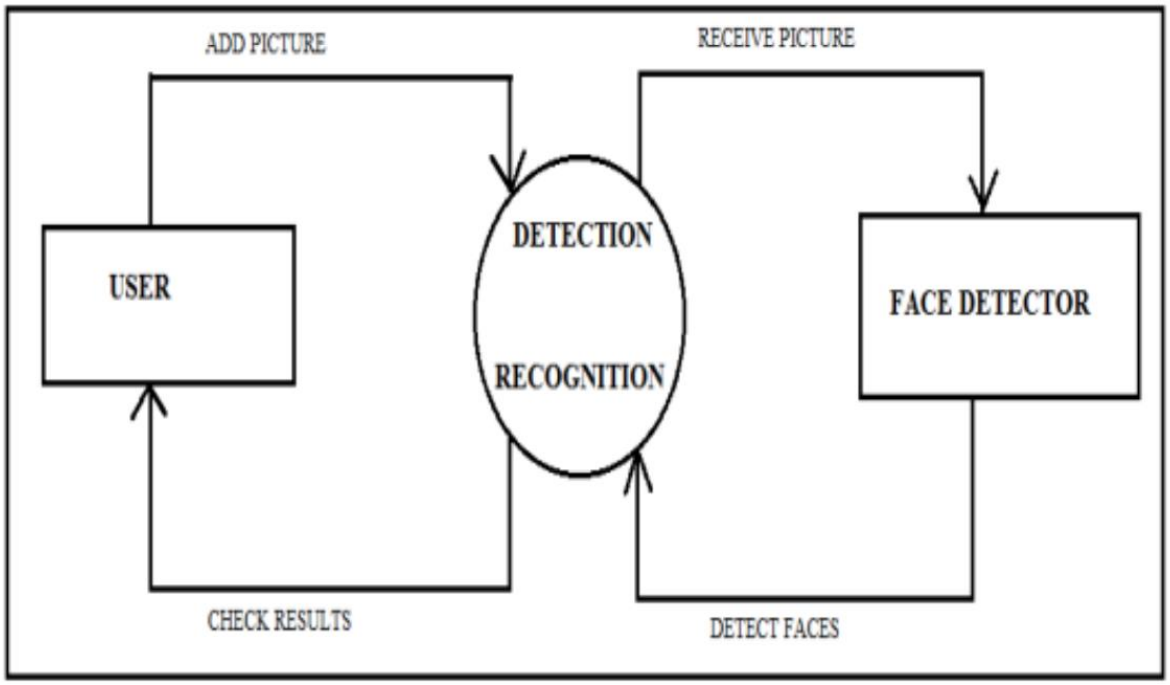
Level – 2


Start
Input face
No
Is Skin Present?
Yes
No
Is Human Face features Present?
Yes
Load Faces from Training Directory
Face match Algorithm
No
Is Face Matched?
Yes
Retrieve age from database
Display age
Stop

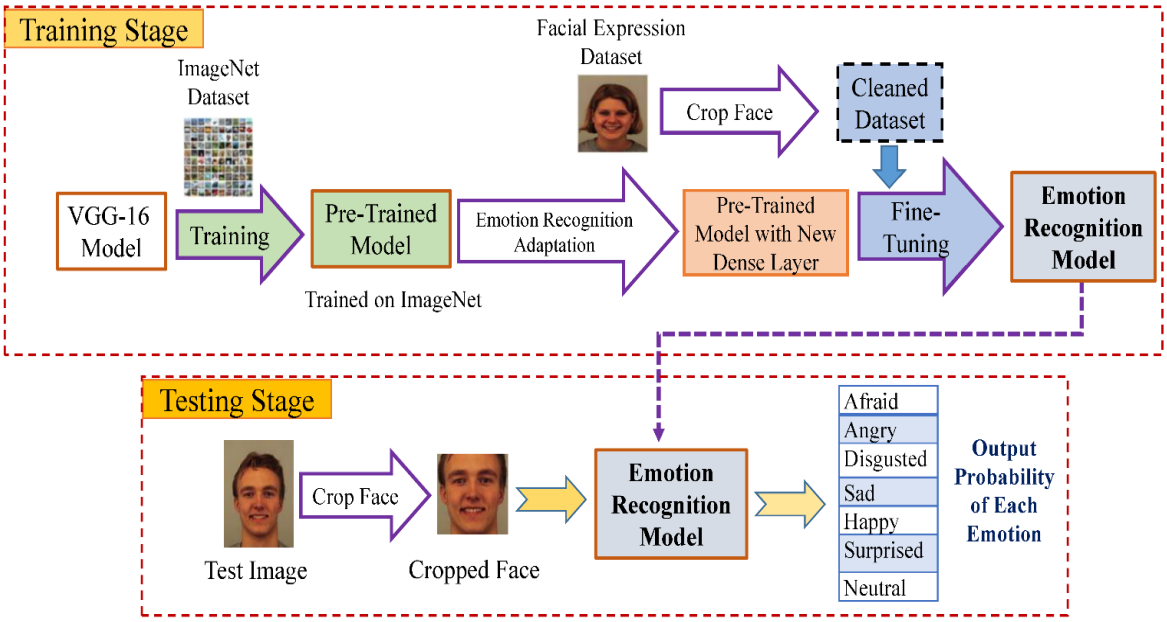
Class Diagram



User Case Diagram



Face Emotion Recognition



CHAPTER 4

SYSTEM REQUIREMENT

HARDWARE SPECIFICATION

- Processor : Dual Core
- RAM : 2GB
- Hard Disk : 160GB

SOFTWARE SPECIFICATION

- Front End : PYTHON 3.6

PACKAGE USED

- Tensorflow
- Scikit-learn
- Tkinter
- Numpy
- Matplotlib
- Pillow
- ipython

ABOUT SOFTWARE

Python

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. User do not need to compile ymy program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − User can actually sit at a Python prompt and interact with the interpreter directly to write ymy programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Programming Language

1. Python is currently the most widely used multi-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
5. The biggest strength of Python is huge collection of standard library which can be used for the following:
    - Machine Learning
    - GUI Applications (like Kivy, Tkinter, PyQt etc.)
    - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
    - Image processing (like OpenCV, Pillow)
    - Web scraping (like Scrapy, BeautifulSoup, Selenium)
    - Test frameworks
    - Multimedia
    - Scientific computing
    - Text processing

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − User can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Packages

TensorFlow

Image recognition consists of pixel and pattern matching to identify the image and its parts. Each image is a container of pixels which in turn are the combination of numbers. These numbers represent the color depth. Here the character recognition can be executed by this TensorFlow framework.

Scikit-learn

It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Tkinter

Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application. Here the application is designed with button to upload the input image.

NumPy

NuAmPy can be used to perform a wide variety of mathematical operations on arrays. The comparision and prediction with the dataset can be implemented by using various mathematical calculation in recognizing the handwritten character input.

Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

Pillow

Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images.

ipython

It is an interactive command-line terminal for Python. IPython offers an enhanced read-eval-print loop (REPL) environment particularly well adapted to scientific computing.

NEURAL NETWORKS

Neural network is a system inspired by human brain function, consists of neurons connected in parallel with the ability to learn. A basic design of neural network has input layer, hidden layer, and output layer. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract pattern and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been to analyze. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining.

NETWORK LAYERS

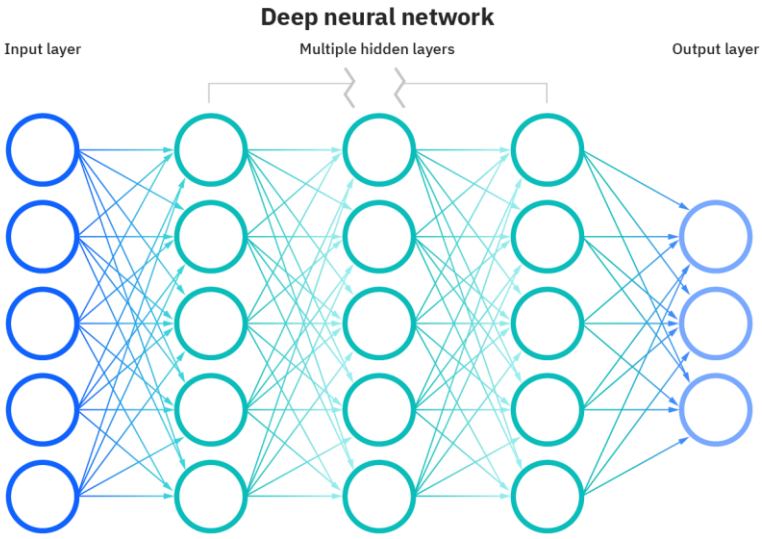
The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units.

• The activity of the input units represents the raw information that is fed into the network.

• The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

• The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and the hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.



CONVOLUTIONAL NEURAL NETWORKS

A special type Neural Networks that works in the same way of a regular neural network except that it has a convolution layer at the beginning. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and softmax layers.

About Algorithm

Convolutional Neural Networks

CNN are made up of a large number of interconnected neurons that have learnable weights and biases. In the architecture of CNN the neurons are organized as layers. It consist of an input layer, many hidden layers and an output layer. If the network has a large number of hidden layers the same are generally referred as deep neural networks. The neurons in the hidden layers of CNN are connected to a small region (receptive field) of the input space generated from the previous layer instead of connecting to all, as in the fully connected network like Multi Layered Perceptron (MLP) networks. This approach reduces the number of connection weights (parameters) in CNN compared to MLP. As a consequence, CNN takes less time to train for the networks of similar size. The input to the typical CNN are two dimensions (2D) array of data such as images. Unlike the regular neural network the layers of a CNN are arranged in three dimensions (width, height and depth).

The followings are the type of layers which as commonly found in the CNNs.

- Input Layer is basically a buffer to hold the input and pass it on to the next layer.
- Convolution Layer performs the core operation of feature extraction. It performs Convolution operation of the input data. The convolution operation is executed by sliding the kernel over the input, and performs sum of the product at every location. The step size with which the kernel slides are known as stride. Numerous convolution operations are performed on the input by using different kernel, which results in different feature maps, the number of feature map produced in a convolutional layer is also known as the depth of the layer.
- Rectified Linear Unit (ReLU) is an activation function used to introduce non linearity. It replaces negative value with zero, which can speed up the learning process. The output of every convolution layer is passed through the activation function.
- Pooling layer reduces the spatial size of each feature map, which in turn reduces the computation in the network. Pooling also uses a sliding window that moves in stride across the feature map and transform it into representative values. Min pooling, average pooling and max pooling are commonly used.
- Fully connected layer connects every neuron in the layer to all the neurons in the previous layer. It learns the non-linear combination of the features and used for classifying or predicting the output. For

classification problems, the fully connected layer is generally followed by a soft-max layer, it produces the probability of each class for the given input. And for regression problems, it is followed by a regression layer to predict the output.

The architecture of the CNN is organized according to the problem to be addressed. Like any other neural networks the CNN is intelligent and they can learn. Learning is achieved through training (supervised). The CNN are feedforward networks and uses back-propagation training algorithm. The training is performed in two passes; forward and backward pass. In the forward pass the network weight and bias are initialized with small random numbers and compute the network output by using training input. The error is computed by comparing the network output with the desired training output. In the backward pass the error propagates backward and all the weights and bias are adjusted to minimize the error. The process is repeated until the desired result is obtained. Once the network is trained with a suitable dataset, it can be used for solving a specific problem.

Many researches have developed different CNN architectures for classification. Some of the most popular deep CNN networks include AlexNet, ZfNet and LeNet.

AlexNet

AlexNet accepts input of size 227 × 227 × 3 and consist of 25 layers. It has 5 convolution layers which are followed by ReLU layer and max pooling layer. The first two convolution layers are also followed by crosschannel normalization layer. Cross channel normalization carries out channel wise normalization. It replaces each element with a normalized value obtained from neighbouring cells. It also has three fully connected layers followed by ReLU Layer and the first two fully connected layer is also followed by dropout layers. The output of the last fully connected layer is given as an input to softmax which produces the probability distribution of 1000 classes. The AlexNet was designed for recognizing objects in ImageNet

ZfNet

ZfNet has similar architecture to AlexNet but differs in the filter size and the number of filters used. In the first convolution layer, the AlexNet uses filter of size 11×11 with a stride of 4 and the ZfNet uses 7×7 and stride of 2. The ZfNet uses 512,1024 and 512 filters in the third,fourth and fifth convolution layer whereas the AlexNet uses 384,384 and 256 filters respectively. The ZfNet was also designed for recognizing objects in ImageNet.

LeNet

LeNet accepts a grayscale image of size 32×32 as an input. The architecture consists of two sets of convolutional and average pooling layers, followed by a flattening convolutional layer, then follows two fully connected layers and a softmax layer for classification. The LeNet is a pioneering work in recognizing hand written digits from the images by using CNN.

CHAPTER 5

SYSTEM IMPLEMENTATION

Automatic gender classification and age detection may be a fundamental task in computer vision, which has recently attracted immense attention. It plays a very important role in an exceedingly wide selection of the real-world applications like targeted advertisement, forensic science, visual surveillance, content-based searching, human computer interaction systems, etc. for instance I are able to use this method to display advertisement supported different gender and different age bracket. This method may be employed in different mobile applications where there's some age restricted content in order that only appropriate user can see this content. However, gender classification and age approximation is still a difficult task. I propose a model which can first perform feature extraction on the input image which can classify eyes, lips, beard, hair, etc. Supported these features the model will classify the gender as male or female. We've used Haar Cascade for feature extraction purpose. Age is estimated with the assistance of Caffe Model. The age classifier takes an image of an individual's face of size 256x256 as an input to the algorithm that's then cropped to 227x227. The age classifier returns a integer representing the age range of the individual. There are 8 possible age ranges , that the age classifier returns an integer between 0 and seven. The gender classifier returns a binary result where 1 indicates male and 0 represents female.

Module Description

Gender Classification

Images may not be perfect. There are many noises which are redundant. This can decrease system performance. To increase accuracy rate I have to make proper and effective feature extraction. This can be global or local which depends on shape, color, orientation.

1) Edge detection

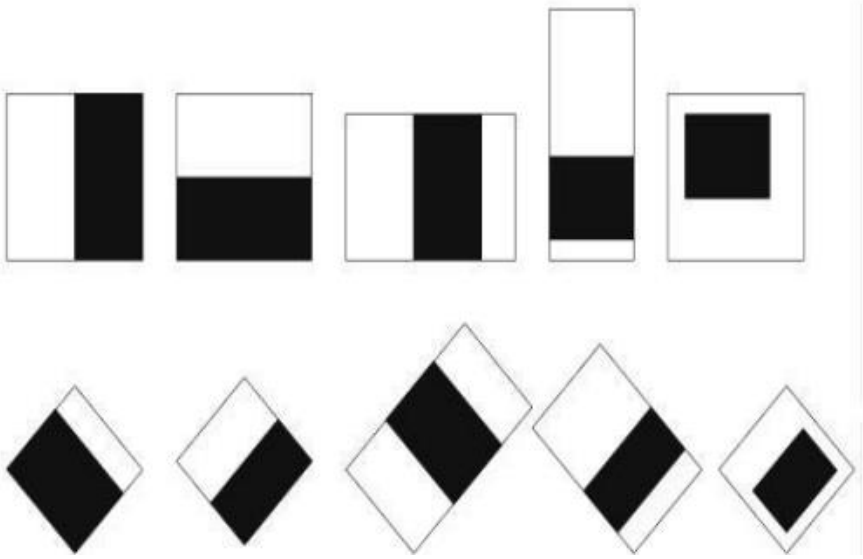
Edge feature is mostly used for detecting the object. It finds the discontinuities in gray level. I can say that edge is the boundary between the regions.

2) Haar– like features:

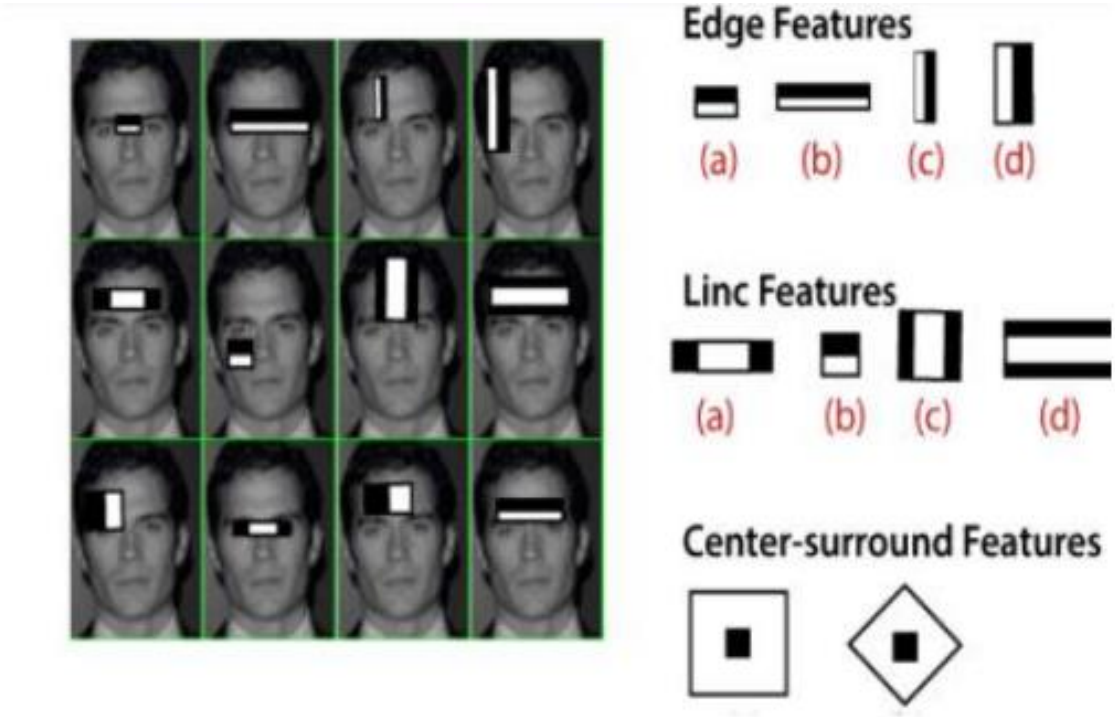
Viola and Jones proposed an algorithm which is called Haar-Classifiers for rapid object detection and pedestrian detection is applied. it is done with the haar like features which can be calculated efficiently by using Adaboost classifier and integral images in cascade classifier. Haar-like features can have high accuracy and in low cost. Haar cascade is mostly used for face detection because of its easy calculation.

3) Detector using haar -Like features

In face detection, the image is first scanned, looking for patterns with indicate the presence of a face in the image. This is done by using haarlike features. The haar like features are created by two or three adjacent rectangles with different contrast values.



HAAR FEATURES



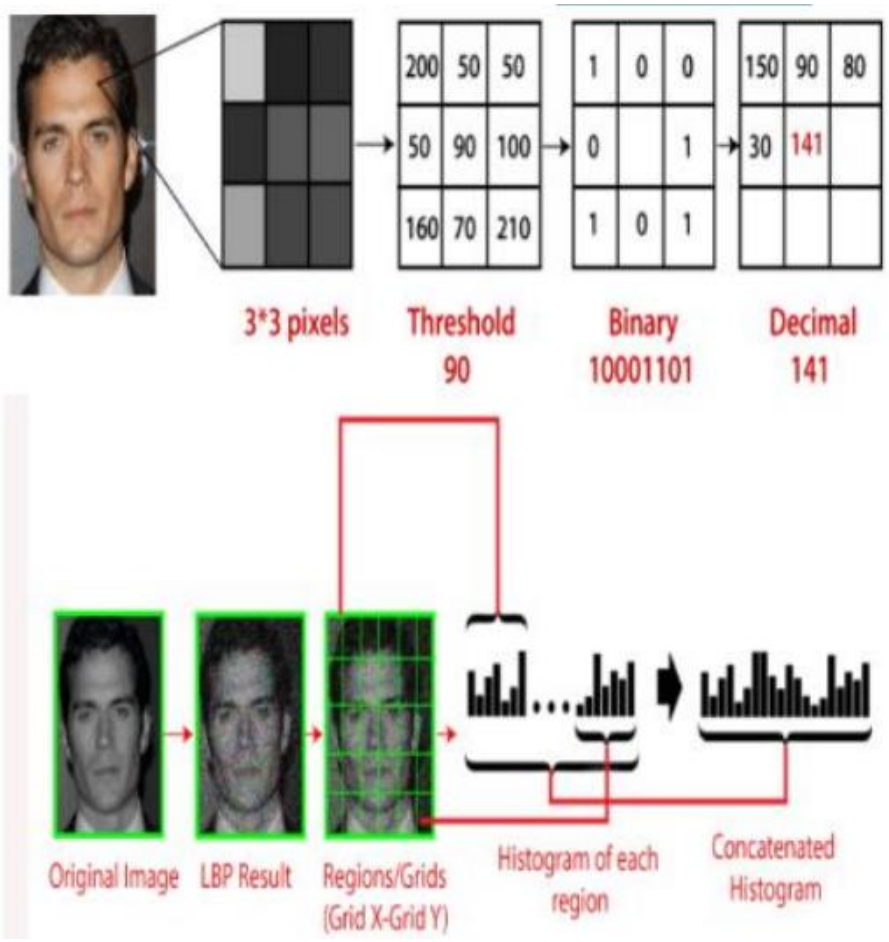
Working of HAAR Features

There are separate intensity values of black and white pixels which gives dark and light regions. Any object can be detected by using haar like features. I just have to adjust the size of rectangle that I can scale up and down the image.

4)Texture feature

This feature was proposed by Robert M. Haralick in 1973. These features make use of the statistics which summarize the relative frequency distribution which describes how one gray tone is spatially related to the graytone. Local Binary Pattern Algorithm is used for extracting texture features.

This is simple but effective algorithm to extract texture features. Using LBP computation an intermediate image is generated which describes the original image in specific way. For concept of sliding window the parameters like radius and neighbors are used. The input facial image is grayscale . This approach get the block of this image as 3x3 matrix. The 3x3 matrix contains the intensity of each pixel (0-255). I take the central value of the matrix and use it as threshold. This value is used to define the new values of the matrix. Each neighbor of the central value is set as a new binary value. The value is set to 1 if it is greater than the threshold value and 0 if the value is lower than the threshold value. The matrix now contains only binary values. I now concatenate each binary value line by line or clockwise but the out will be same. Then convert the binary value into decimal value. Likewise each pixel in the matrix is converted into decimal value. The resultant image represents better characteristics of original image. Histograms are derived from each such matrix of image and all the histograms are concatenated which show the better characteristics of the original image.



Histogram Generation

Age Classification

Steps for feature extraction:

1. An input image or class of images.

2. Pre-trained model(.caffe-model)-binary file which has the weights, gradients and biases for each layer of the network.

3. Model definition (.prototxt) file which has the network structure that is used.

4. Target feature extraction layer(eg:-“fc7” in Alexnet)

Convolution Neural Network

The main building block of CNN is the convolutional layer. Convolution is a mathematical operation to merge two sets of information. In my case the convolution is applied on the input data using a convolution filter to produce a feature map. There are a lot of terms being used so let's visualize them one by one. On the left side is the input to the convolution layer, for example the input image. On the right is the convolution filter, also called the kernel, I will use these terms interchangeably. This is called a 3x3 convolution due to the shape of the filter. I perform the convolution operation by sliding this filter over the input. At every location, I do element-wise matrix multiplication and sum the result. This sum goes into the feature map. The green area where the convolution operation takes place is called the receptive field. Due to the size of the filter the receptive field is also 3x3.

Caffe for Age approximation

Caffe is a CNN framework which allows researchers and other practitioners to build a complex neural network and train it without need to write much code. For estimation of age using the convolution neural network, gathering a large dataset for training the algorithm is a tedious and time consuming job. The dataset needs to be well labeled and from social image database which has the private information of the subjects i.e. age.

The Adience Dataset

The benchmark for this problem, uses the Adience dataset which is composed of images scraped from Flickr.com albums that were labeled for age and gender. The benchmark uses 8 classes for age groups (0–2, 4–6, 8–13, 15–20, 25–32, 38–43, 48–53, 60+), and therefore I treated both gender prediction and age prediction as a classification problem. The Adience dataset is relatively small (containing 34,795 images), so I also used the IMDB+Wiki dataset which is the largest dataset publicly available for age and gender (containing 523,051 face images).

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	Total
Male	745	928	934	734	2308	1294	392	442	8192
Female	682	1234	1360	919	2589	1056	433	427	9411
Both	1427	2162	2294	1653	4897	2350	825	869	19487

After the primary 2 pooling layers, there are local response normalization (LRN) layers. LRN could be a technique that was first introduced in as the way to assist the generalization of deep CNNs. The idea behind it's to introduce lateral inhibition between the various filters in a very given convolution by making them "compete" for big activations over a given segment of their input. This effectively prevents repeated recording of the identical information in slightly different forms between various kernels watching the identical input area and instead encourages fewer, more prominent, activations in some for a given area. If a(x,y) is that the activation of a neuron by applying kernel i at position (x, y), then it's local response normalized activation b(x,y) is given by

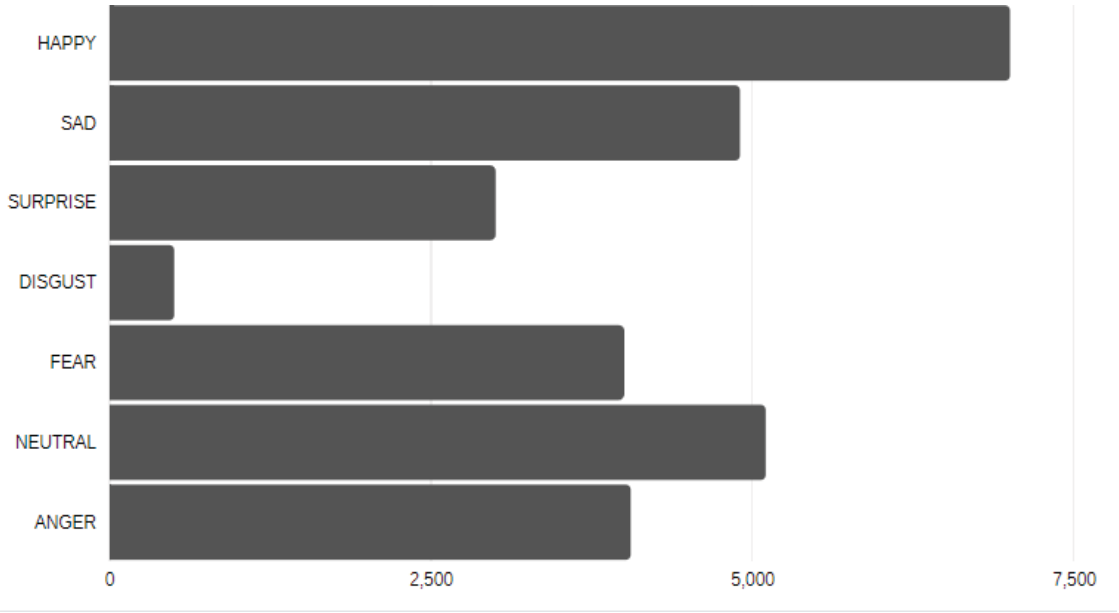
$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

where k,n,α, and β are all hyper-parameters. The parameter n is that the number of "adjacent" kernel maps (filters) over which the LRN is run, and N is that the total number of kernels therein given layer.

Face Recognition

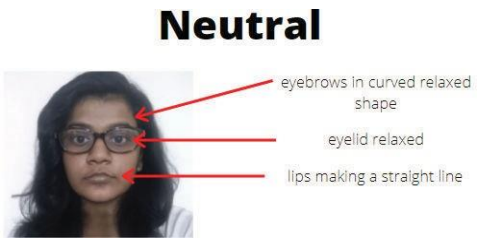
Data Collection

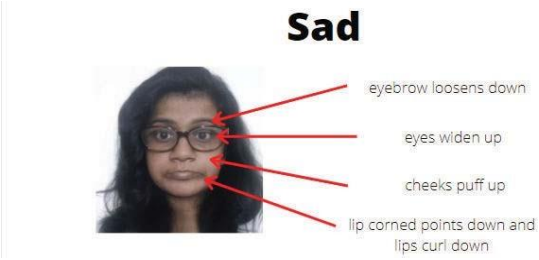
There were enumerable datasets available on emotion detection, I analyzed a lot of them and decided on working with the fer2013 challenge data set as it is a cleaned dataset with 28,709 sets that can be used to train the model, moreover, it covers the seven universal emotions, therefore, making it the ideal dataset



Finding Insights

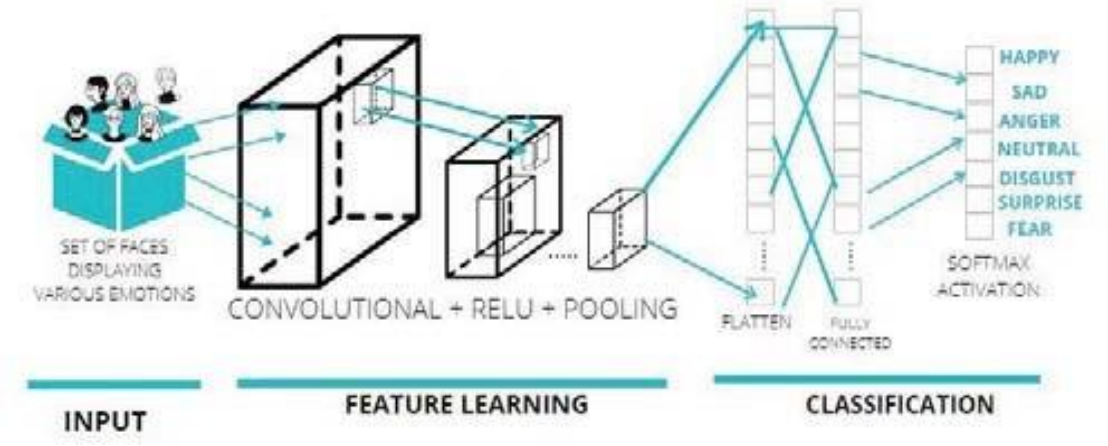
Finding insights is one of the most important parts of deciding the algorithm, these pictures help us identify which characteristics help us figure out if the expression belongs to Happy, sad, fear, neutral, angry, surprise, or disgust.





Applying Algorithm

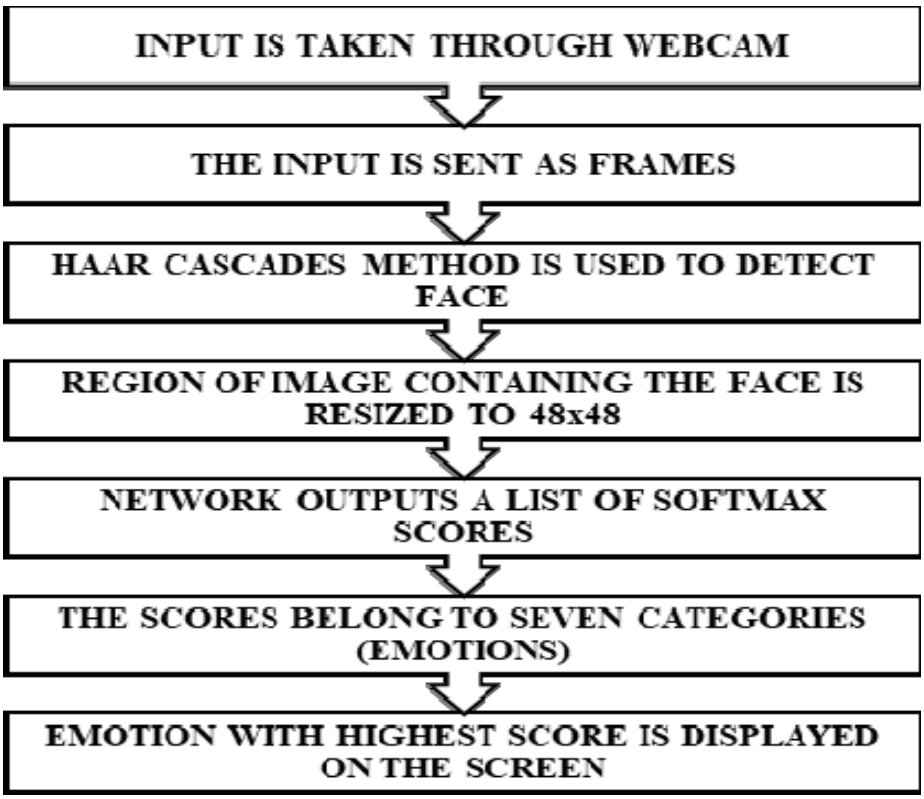
A CNN is a Deep Learning algorithm which is highly efficient for visual models, it learns the differences and helps us from differentiating images from one another. It is one of the fastest and most efficient classification algorithms. The factor that separates CNN from other algorithms is its learning capabilities which is a derivative of the human brain's neural network. It consists of an input and output layer connected by various mathematical models. CNN tends to replicate the behavior of the human occipital cortex. The CNN used in the project is Three Layer Depth Wise separable Convolutional neural network, the global average pooling and batch normalization, and the softmax scoring, the Conv2D are the Different Layers of the applied algorithm.



Training the Model

- Depth-wise separable convolutions composing two different layers: depth-wise convolutions and pointwise convolutions. The main purpose of these layers is to separate the spatial cross-correlations from the channel cross-correlations. They do this by first applying a D×D filter on every M input channels and then applying N1×1×M convolution filters to combine the M input channels into N output channels. Applying 1× 1 × M convolutions combines each value in the feature without considering their spatial relationships within the channel.
- The testing dataset of fer2013 includes 3589 images that belong to one of the seven emotions, the testing dataset is then inputted to the trained model, further ahead, the expected output and the received output is further compared to calculate the accuracy of the trained model depending upon the correctness.

Emotion Prediction



Conclusion

The project explains significant approach to gender recognition problem along with different approach for age approximation. This study was the trial of this idea. Although the results aren't perfect, they're promising to future studies. My next step is to make a higher Haar cascade and use this method for other multiclass problems. Training the haar cascade classifier with much more data can surely improve the accuracy of the classifier. The easy availability of big image collections provides modern machine learning based systems with effectively endless training data, though this data isn't always suitably labeled for supervised learning. The system was programmed in python language. Both real time and static face detection was carried out. Taking example from the related problem of face recognition I explore how well deep CNN perform on these tasks using Internet data. I offer results with a lean deep-learning architecture designed to avoid over fitting because of the limitation of limited labeled data. The most difficult portion of

this project was fitting the training infrastructure to properly divide the info into folds, train each classifier, cross-validate, and mix the resulting classifiers into a test-ready classifier. I foresee future directions building off of this work to incorporate using gender and age classification to help face recognition, improve experiences with photos on social media, and far more. Finally I hope that additional data training will improve the algorithm to provide better results.

BIBLOGRAPHY

Referred Books

"Automate The Boring Stuff With Python, 2Nd Edition: Practical Programming For Total Beginner", SL Swegart, 2020.

"Learning Python 5ed: Powerful Object-Oriented Programming" O'Relly – 2014.

"Elements of Programming Interviews in Python: The Insiders' Guide", Adan Aziz, Tsung-Hsen Lee – 2021.

Referred websites

https://www.tensorflow.org/about

https://en.wikipedia.org/wiki/TensorFlow

https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html

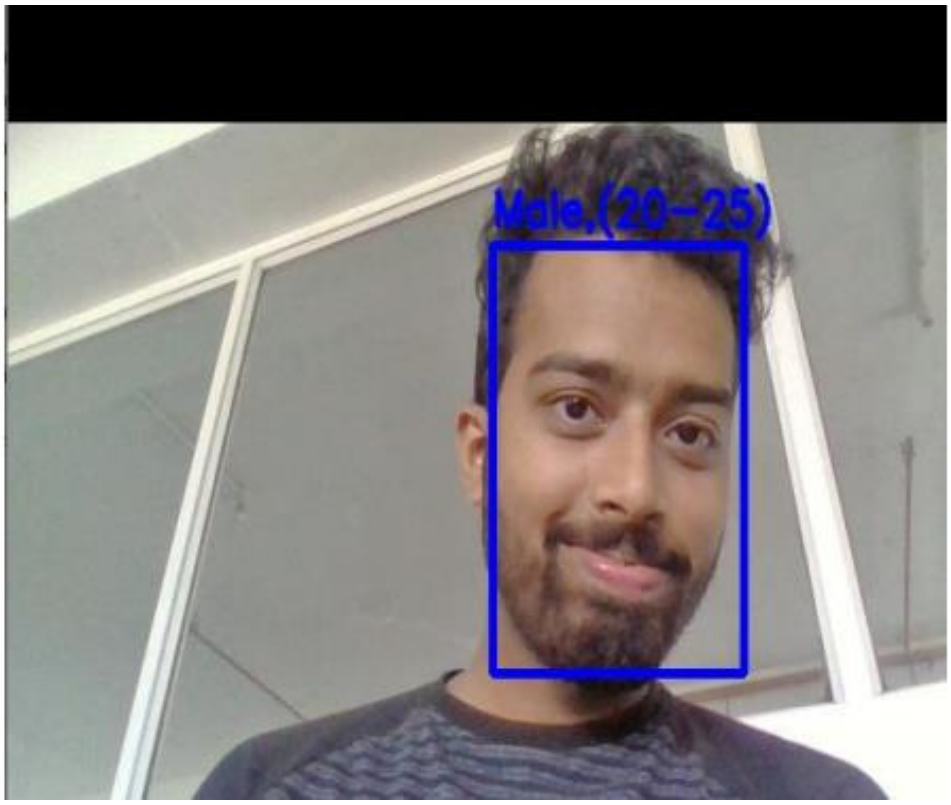
https://www.python.org/about/

https://www.w3schools.com/python/python_intro.asp

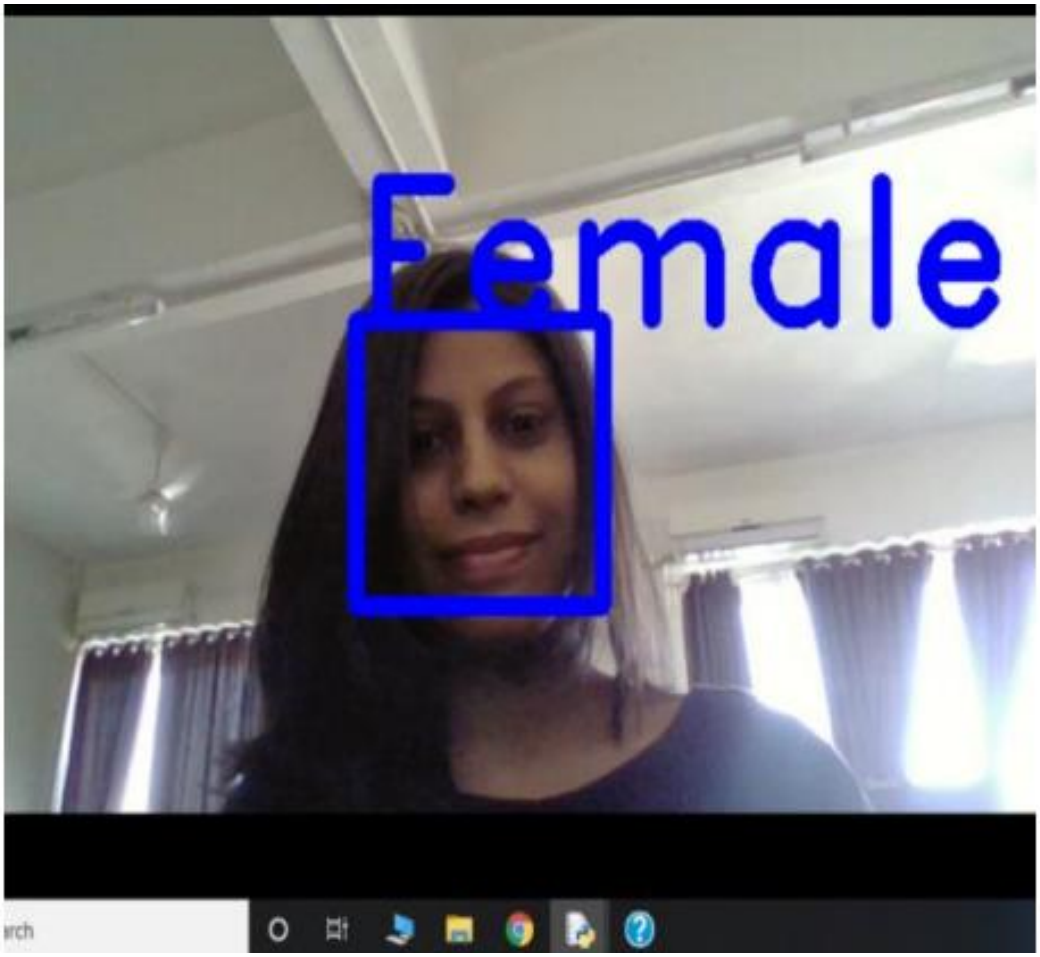
https://www.geeksforgeeks.org/python-programming-language/

SCREEN SHOTS

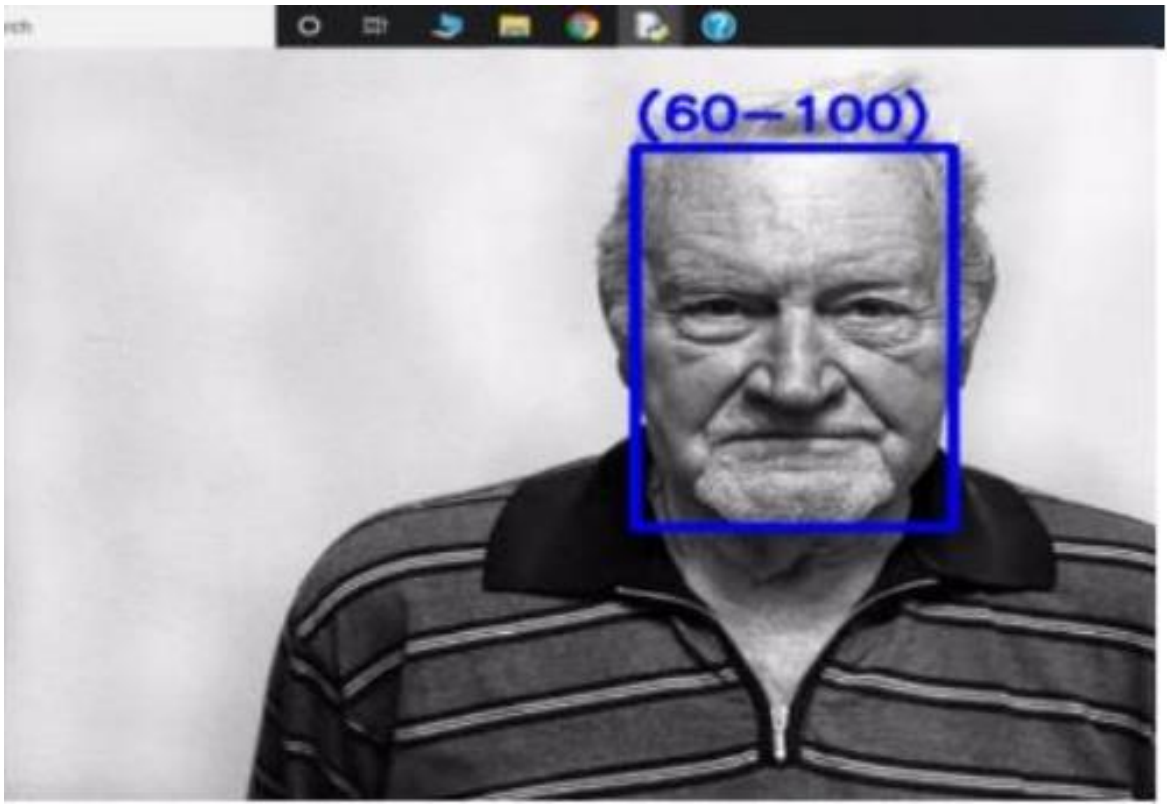
Age Prediction



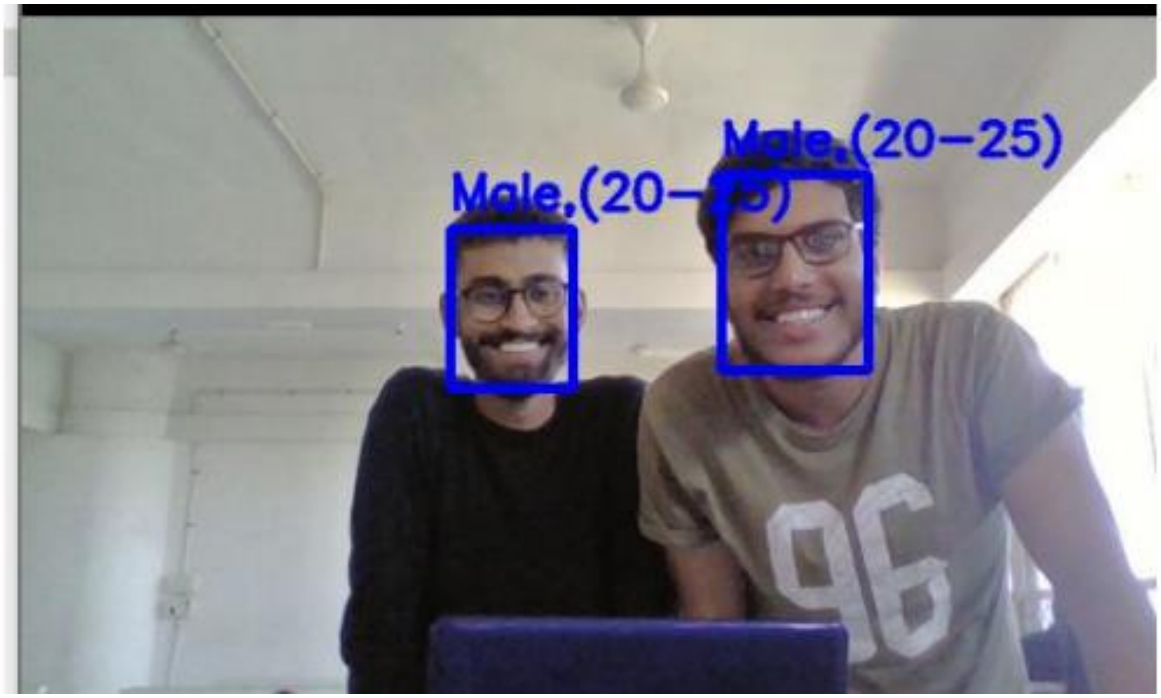
Gender Prediction



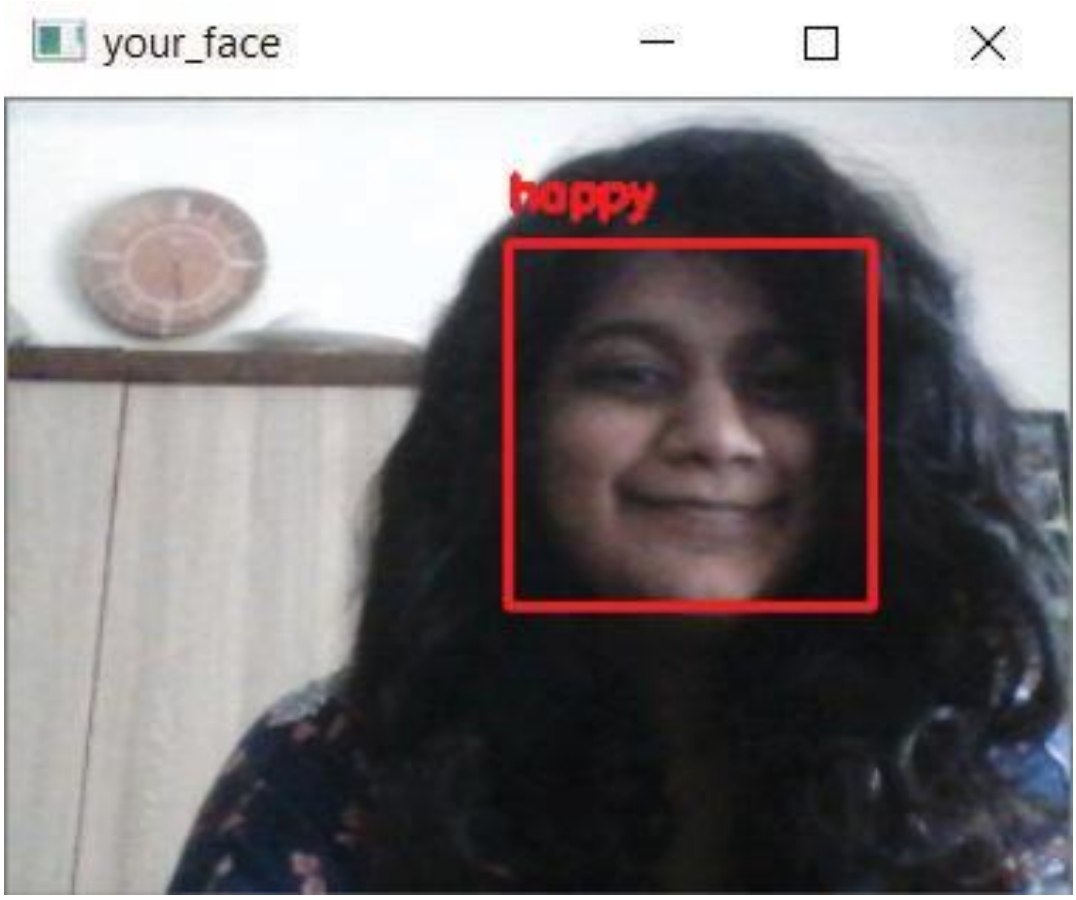
Age Group Prediction



Age and Gender Prediction



Emotion Prediction



Sample Code

Camera Capture

```
import cv2
video_capture = cv2.VideoCapture(0)
faceCascade = cv2.CascadeClassifier('style.xml')
video_capture.set(3, 480) #set width of the frame
video_capture.set(4, 640) #set height of the frame
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
age_list = ['(0, 2)', '(4, 6)', '(8, 12)', '(15, 20)', '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)']
gender_list = ['Male', 'Female']

def load_caffe_models():
age_net = cv2.dnn.readNetFromCaffe('deploy_age.prototxt', 'age_net.caffemodel')
gender_net = cv2.dnn.readNetFromCaffe('deploy_gender.prototxt', 'gender_net.caffemodel')
return(age_net, gender_net)

def video_detector(age_net,gender_net):
font = cv2.FONT_HERSHEY_SIMPLEX #Type of font
while True:
check,frame = video_capture.read()
frame=cv2.flip(frame,1)
#converted our Webcam feed to Grayscale.**< Most of the operations in OpenCV are done in grayscale>**
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(gray, 1.2,10,cv2.CASCADE_SCALE_IMAGE,(30,30))
print(check)
print(frame)
# Draw a rectangle around the faces
```

```
for (x, y, w, h) in faces:
#Make rectangle on face
#print(x,y,w,h)#print The coordinates of face
cv2.rectangle(frame, (x, y), (x + w, y + h), (255,255,0), 2)
# Get Face as Matrix and copy it
face_img = frame[y:y + h, h:h + w].copy()
print(face_img)
blob=cv2.dnn.blobFromImage(face_img,1,(244,244),MODEL_MEAN_VALUES,swapRB=True)#**
# Predict Gender
gender_net.setInput(blob)
gender_preds = gender_net.forward()
gender = gender_list[gender_preds[0].argmax()]
# Predict Age
age_net.setInput(blob)
age_preds = age_net.forward()
age = age_list[age_preds[0].argmax()]
overlay_text = "%s %s" % (gender, age)
cv2.putText(frame, overlay_text, (x, y), font, 1, (255, 255, 255), 2, cv2.LINE_AA)
cv2.imshow('frame', frame)
key=cv2.waitKey(1)
if key == 27:
break
video_capture.release()
cv2.destroyAllWindows()

def main():
age_net, gender_net = load_caffe_models()  # load caffe models (age & gender)
video_detector(age_net, gender_net)  # prediction age & gender

if __name__ == "__main__":
main()
```

Face Detection

```
# Import the necessary libraries
import numpy as np
import cv2
import matplotlib.pyplot as plt
get_ipython().magic('matplotlib inline')
#  Loading the image to be tested
test_image = cv2.imread('baby1.jpg')
# Converting to grayscale as opencv expects detector takes in input gray scale images
test_image_gray = cv2.cvtColor(test_image, cv2.COLOR_BGR2GRAY)
# Displaying grayscale image
plt.imshow(test_image_gray, cmap='gray')
# Since we know that OpenCV loads an image in BGR format so we need to convert it into RBG format to be able to display its true colours. Let us write a small function for that.
def convertToRGB(image):
return cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# # Haar cascade files
# Loading the classifier for frontal face
haar_cascade_face = cv2.CascadeClassifier('data/haarcascades/haarcascade_frontalface_alt2.xml')
# # Face detection
faces_rects = haar_cascade_face.detectMultiScale(test_image_gray, scaleFactor = 1.2, minNeighbors = 5);
# Let us print the no. of faces found
print('Faces found: ', len(faces_rects))
# Our next step is to loop over all the co-ordinates it returned and draw rectangles around them using Open CV.We will be drawing a green rectangle with thicknessof 2
for (x,y,w,h) in faces_rects:
cv2.rectangle(test_image, (x, y), (x+w, y+h), (0, 255, 0), 2)
# Finally, we shall display the original image in coloured to see if the face has been detected correctly or not.
#convert image to RGB and show image
plt.imshow(convertToRGB(test_image))
#  Let us create a generalised function for the entire face detection process.
```

```
def detect_faces(cascade, test_image, scaleFactor = 1.1):
# create a copy of the image to prevent any changes to the original one.
image_copy = test_image.copy()
#convert the test image to gray scale as opencv face detector expects gray images
gray_image = cv2.cvtColor(image_copy, cv2.COLOR_BGR2GRAY)
# Applying the haar classifier to detect faces
faces_rect = cascade.detectMultiScale(gray_image, scaleFactor=scaleFactor, minNeighbors = 5)
for (x, y, w, h) in faces_rect:
cv2.rectangle(image_copy, (x, y), (x+w, y+h), (0, 255, 0), 15)
return image_copy
# Testing the function on new image
#loading image
test_image2 = cv2.imread('group.jpg')
#call the function to detect faces
faces = detect_faces(haar_cascade_face, test_image2)

#convert to RGB and display image
plt.imshow(convertToRGB(faces))

# Saving the final image

cv2.imwrite('image1.png',faces)
```

Detect Age Group

```
#A Gender and Age Detection program by Mahesh Sawant

import cv2
import math
import argparse
```

```
def highlightFace(net, frame, conf_threshold=0.7):
frameOpencvDnn=frame.copy()
frameHeight=frameOpencvDnn.shape[0]
frameWidth=frameOpencvDnn.shape[1]
blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

net.setInput(blob)
detections=net.forward()
faceBoxes=[]
for i in range(detections.shape[2]):
confidence=detections[0,0,i,2]
if confidence>conf_threshold:
x1=int(detections[0,0,i,3]*frameWidth)
y1=int(detections[0,0,i,4]*frameHeight)
x2=int(detections[0,0,i,5]*frameWidth)
y2=int(detections[0,0,i,6]*frameHeight)
faceBoxes.append([x1,y1,x2,y2])
cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
return frameOpencvDnn,faceBoxes

parser=argparse.ArgumentParser()
parser.add_argument('--image')
args=parser.parse_args()
faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"
```

```
MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']

faceNet=cv2.dnn.readNet(faceModel,faceProto)
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)
video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0 :
hasFrame,frame=video.read()
if not hasFrame:
cv2.waitKey()
break
resultImg,faceBoxes=highlightFace(faceNet,frame)
if not faceBoxes:
print("No face detected")
for faceBox in faceBoxes:
face=frame[max(0,faceBox[1]-padding):
min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
:min(faceBox[2]+padding, frame.shape[1]-1)]

blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES, swapRB=False)
genderNet.setInput(blob)
genderPreds=genderNet.forward()
gender=genderList[genderPreds[0].argmax()]
print(f'Gender: {gender}')

ageNet.setInput(blob)
agePreds=ageNet.forward()
age=ageList[agePreds[0].argmax()]
```

```
print(f'Age: {age[1:-1]} years')

cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
cv2.imshow("Detecting age and gender", resultImg)
```

Age and Gender Label Fixing

```
import cv2
video_capture = cv2.VideoCapture(0)
faceCascade = cv2.CascadeClassifier('style.xml')
video_capture.set(3, 480) #set width of the frame
video_capture.set(4, 640) #set height of the frame
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
age_list = ['(0, 2)', '(4, 6)', '(8, 12)', '(15, 20)', '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)']
gender_list = ['Male', 'Female']
def load_caffe_models():
age_net = cv2.dnn.readNetFromCaffe('deploy_age.prototxt', 'age_net.caffemodel')
gender_net = cv2.dnn.readNetFromCaffe('deploy_gender.prototxt', 'gender_net.caffemodel')
return(age_net, gender_net)
def video_detector(age_net,gender_net):
font = cv2.FONT_HERSHEY_SIMPLEX #Type of font
while True:
check,frame = video_capture.read()
frame=cv2.flip(frame,1)
#converted our Webcam feed to Grayscale.**< Most of the operations in OpenCV are done in grayscale>**
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(gray, 1.2,10,cv2.CASCADE_SCALE_IMAGE,(30,30))
print(check)
print(frame)
# Draw a rectangle around the faces
```

```
for (x, y, w, h) in faces:
#Make rectangle on face
#print(x,y,w,h)#print The coordinates of face
cv2.rectangle(frame, (x, y), (x + w, y + h), (255,255,0), 2)
# Get Face as Matrix and copy it
face_img = frame[y:y + h, h:h + w].copy()
print(face_img)
blob=cv2.dnn.blobFromImage(face_img,1,(244,244),MODEL_MEAN_VALUES,swapRB=True)#**
# Predict Gender
gender_net.setInput(blob)
gender_preds = gender_net.forward()
gender = gender_list[gender_preds[0].argmax()]
# Predict Age
age_net.setInput(blob)
age_preds = age_net.forward()
age = age_list[age_preds[0].argmax()]
overlay_text = "%s %s" % (gender, age)
cv2.putText(frame, overlay_text, (x, y), font, 1, (255, 255, 255), 2, cv2.LINE_AA)
cv2.imshow('frame', frame)
key=cv2.waitKey(1)
if key == 27:
break
video_capture.release()
cv2.destroyAllWindows()
def main():
age_net, gender_net = load_caffe_models()  # load caffe models (age & gender)
video_detector(age_net, gender_net)  # prediction age & gender
if __name__ == "__main__":
main()
```

CONCLUSION:

Recently there has been some activity in designing invariant recognition methods which do not require invariant features analysis, recognition-by-components theory, bottom-up and top-down processing, and Fourier analysis .In future in order to achieve better accurate in both identity and verification tasks of pattern recognition in various research we can use some adaptive hybrid machine learning technique. These include Statistical Techniques, Structural Techniques, Template Matching, Neural Network Approach, Fuzzy Model and Hybrid Models. the performance of the face recognition algorithms under occlusion is in general poor. The face may be occluded by other objects in the scene or by sunglasses or other things. Occlusion may be unintentional or intentional. Facial recognition system have been associated generally with very costly top secure application. Today the core technologies have evolved and the cost of equipment is going down. Certain applications of face recognition technology are now cost effective, reliable and highly accurate.